frivol

Generated by Doxygen 1.8.1.2

Tue May 14 2013 01:11:13

Contents

1 Class Index **1**

1.1 Class List 1

2 Class Documentation **3**

2.1 frivol::Array< T > Class Template Reference 3

2.1.1 Detailed Description 3

2.1.2 Constructor & Destructor Documentation 3

2.1.2.1 Array 3

2.1.3 Member Function Documentation 3

2.1.3.1 operator[] 3

2.1.3.2 operator[] 4

2.2 frivol::PriorityQueueConcept< X, PriorityT > Class Template Reference 4

2.2.1 Detailed Description 4

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

frivol::Array< T >
Simple fixed-size array of elements of type T 3
frivol::PriorityQueueConcept< X, PriorityT > 4

Chapter 2

Class Documentation

2.1 frivol::Array< T > Class Template Reference

Simple fixed-size array of elements of type T.

`#include <array.hpp>`

Public Member Functions

- Array (Idx size)
- Idx getSize () const

  Returns the size of the array.

- const T & operator[] (Idx index) const
- T & operator[] (Idx index)

2.1.1 Detailed Description

template<typename T>class frivol::Array< T >

Simple fixed-size array of elements of type T.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 template<typename T > frivol::Array< T >::Array (Idx *size*)

Creates an array with all elements default-constructed.

Parameters

size	The size of the array.

2.1.3 Member Function Documentation

2.1.3.1 template<typename T > const T & frivol::Array< T >::operator[] (Idx *index*) const

Returns reference to an element in the array.

Parameters

index	The zero-based index of the element.

Exceptions

std::out_of_range	if FRIVOL_ARRAY_BOUNDS_CHECKING is defined and 'index' overflows.

2.1.3.2 template<typename T > T & frivol::Array< T >::operator[] (Idx *index*)

Returns reference to an element in the array.

Parameters

index	The zero-based index of the element.

Exceptions

std::out_of_range	if FRIVOL_ARRAY_BOUNDS_CHECKING is defined and 'index' overflows.

The documentation for this class was generated from the following files:

- /home/topi/unison/Asiakirjat/frivol/frivol/array.hpp
- /home/topi/unison/Asiakirjat/frivol/frivol/array_impl.hpp

2.2 frivol::PriorityQueueConcept< X, PriorityT > Class Template Reference

`#include <priority_queue_concept.hpp>`

Public Member Functions

- **BOOST_CONCEPT_ASSERT** ((boost::LessThanComparable< PriorityT >))
- **BOOST_CONCEPT_USAGE** (PriorityQueueConcept)

Public Attributes

- Idx **size**
- Idx **key**
- PriorityT **priority**

2.2.1 Detailed Description

template<typename X, typename PriorityT>class frivol::PriorityQueueConcept< X, PriorityT >

Concept checking class for priority queues X with priority values of type PriorityT (or NIL). Priority queues are initialized with given size, and contain priority values for keys 0, 1, ..., size-1. Initially, all priority values are NIL. C must support the following operations:

- <construct>(Idx size) creates priority queue for keys 0, 1, ..., size-1.
- bool empty() returns true if all keys have NIL priority.
- Idx pop() returns the key with lowest non-NIL priority and sets the priority of that key to NIL.

- void setPriority(Idx key, PriorityT priority) sets the priority value of key 'key' to non-NIL value 'priority'.
- void setPriorityNIL(Idx key) sets the priority value of key 'key' to NIL. X may assume that PriorityT is ordered with <-operator. X may have undefined behavior if supplied keys are out of range or if pop() is called when empty() returns true.

The documentation for this class was generated from the following file:

- /home/topi/unison/Asiakirjat/frivol/frivol/priority_queue_concept.hpp